http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 7 (2): 292-298, 2008
ISSN 1812-5638

Creation 3D Animatable Face Methodology Using Conic Section-Algorithm

[1]Sushma Jaiswal, [2]Sarita Singh Bhadauria and [3]Rakesh Singh Jadon
[1]Department of Computer Application, UIT,
Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal, MP-462036, India
[2]Department of Electronics Engineering,
[3]Department of Computer Application, MITS Gwalior, MP-474005, India

Abstract: This study describes an efficient method and algorithm to make individual faces for animation from possible inputs. Proposed algorithm reconstruct 3D facial model for animation from two projected pictures taken from front and side views or from range data obtained from any available resources. It is based on extracting features on a face in automatic way and modifying a generic model with detected feature points with conic section and pixalization. Then the fine modifications follow if range data is available. The reconstructed 3Dface can be animated immediately with given parameters. Several faces by one methodology applied to different input data to get a final Animatable face are illustrated.

Key words: Parallel projection, reconstruction, modification, 3Dfacial animation, realistic 3D face, conic section

INTRODUCTION

Bledsoe (1966) described the following difficulties in face recognitions: This recognition problem is made difficult by the great variability in head rotation and tilt, lighting intensity and angle, facial expression, aging, etc. Some other attempts at facial recognition by machine have allowed for little or no variability in these quantities. Yet the method of correlation (or pattern matching) of unprocessed optical data, which is often used by some researchers, is certain to fail in cases where the variability is great. In particular, the correlation is very low between two pictures of the same person with two different head rotations.

Irfanoglu *et al.* (2004) proposed 3D Shape-based Face Recognition using Automatically Registered Facial Surfaces, in this method authors address the use of three dimensional facial shape information for human face identification. Fine registration of facial surfaces is done by first automatically finding important facial landmarks and then, establishing a dense correspondence between points on the facial surface with the help of a 3D face template-aided thin plate spline algorithm. Park *et al.* (2005) proposed 3D Model-Assisted Face Recognition in Video which contains temporal information as well as multiple instances of a face, so it is expected to lead to better face recognition performance compared to still face images. In this study, we introduce a novel face recognition using the geometrical features of normalized 3D face images with the help of conic spline.

The face orientation and illumination causes the face recognization is not very easy in real application. Blanz and Vetter (2003) provide the face recognition method based on 3D morphable model. In this method, a 2D face image is matched to 3D face model which can be morphed according to some parameter. The face recognition can be applied according to those parameters. For any given face image, we should map 2D image to the 3D face models before face recognization. The project used PCA eigenface method to match 2D image to 3D face model.

Automatic face recognition has been a difficult problem in the field of computer vision for many years. Robust face recognition requires the ability to recognize identity despite any variations in appearance the face can have in a scene. There are different types of noses and mouth shapes presents in the human faces i.e., the shape of elliptical, hyperbola and parabola etc. mouth and nose pick point is easy to find using conic section method.

The process proposed here is a sequential procedure that aims to put the face shapes in a standard spatial position. With the help of conic images with the following values for parameter r. $r > 1/2$, $w_1 > 1$ (Hyperbola Section), $r = 1/2$, $w_1 = 1$ (Parabola section), $r < 1/2$, $w_1 < 1$ (Ellipse section), $r = 0$, $w_1 = 0$ (Straight line segment). Conic section gives the standard shape compare to other methods whereas other methods gives spatial rotation techniques and suffer drastic losses in performance when the face is not correctly oriented.

Corresponding Author: Sushma Jaiswal, Department of Computer Application, UIT, Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal, MP-462036, India

NORMALIZATION FACE WITH THE HELP OF CONIC SECTION

There are different types (shapes) of nose presents in the human face i.e., parabola, elliptical and hyperbola etc. depends on conic section. Figure 1 shows the Conic Spline of the nose authentication.

Using the conic spline the refinement of the nose and the algebraic form is as follows; (Fig. 1, 2)

$$f(x, y) = ax^2+2hxy+by^2+2ex+2gy+c = 0 \quad (1)$$

Each conic section in a conic spline is defined by two tangents and one additional point. The knots K_i can be located between the vertices the poly line.

$$K_i = (1-v_i)\, V_i+v_iV_{i+1} \quad (2)$$

where, v_i is the between 0 and 1.
The tangent are defined by the triangle with vertices v_i, V_{i+1} and V_{i+2}.

The additional point is:

$$Z_i = Y_iV_{i+1}+ (1-\gamma)\, K_i+K_{i+1} \quad (3)$$

There are several special cases of the conic section that can be handled in a uniform way by these representations.

- If $v_{i+1} = 0$, then the i^{th} section of the conic spline is the line segment from k_i to V_{i+1}.
- If $v_i = 1$ and $V_{i+1} = 0$ then k_i, k_{i+1} and v_{i+1} collapse to the same point and there is a corner in the sequence of conic sections.

I few use the algorithm of conic spline which represents a conic section using three lines that bound the conic (Fig. 1). The equation of a line is:

$$a_0+a_1x+a_2y = 0 \quad (4)$$

Let the first and last vertices in a polyline be A and B and let point C be an intermediate vertex in the polyline. The first and last vertices are joined by the chord AB.

The guided form of conic is the family of conics is the family of with end points at A and B and tangents AC and BC defined by the equation.

$$(A_0+A_1X+A_2Y)\,(B_0+B_1X+B_2Y) = \rho\,(\mu_0+\mu_1X+\mu_2Y)^2 \quad (5)$$

where, $a_0+a_1x+a_2y = 0$ in Eq. 4 is the line containing the line segment AC,

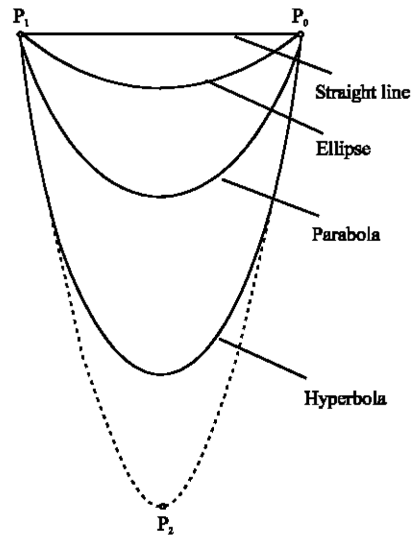

Fig. 1: Conic section of nose bridges-hyperbola, parabola and ellipse section

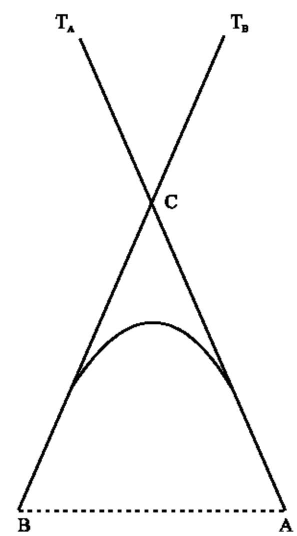

Fig. 2: Conic section

$$b_0+b_1x+b_2y = 0 \quad (6)$$

is the line containing the line segment BC and

$$\mu_0+\mu_1x+\mu_2y = 0 \quad (7)$$

is the line containing the chord AB. The family of conic section is parameterized by P. The value for v in Eq. 1 can be set using the formula:

$$v_i = \frac{A1}{A1+A2} \quad (8)$$

A rational function is simply the ratio of two polynomials. Thus a rational spline is the ratio of two spline functions for example; a-rational b-spline curve can be described with the position vector:

$$\rho(\mu) = \frac{\sum_{k=0}^{n} W_k P_k B_{k,d}}{\sum_{k=0}^{n} W_k B_{k,d}(u)} \tag{9}$$

When the P_k are a sent n+1 control point parameters w_k are weight factors for the control point. The rational B-spline representation is:

$$P(u) = \frac{P_0 B_{0,3} + \left[\frac{r}{(1-r)}\right] P_1 B_{1,3} + P_2 B_{2,3}}{B_{0,3} + \left[\frac{r}{(1-r)}\right] B_{1,3} + B_{2,3}} \tag{10}$$

Then obtain the various conic images with the following values for parameter r. r>1/2, w_1>1 (Hyperbola section), r = 1/2, w_1 = 1 (Parabola section), r<1/2, w_1<1 (Ellipse section), r = 0, w_1 = 0 (Straight line segment).

From this nose pick point, the oriented faces are compensated for three axes, X, Y and Z (Lee *et al*., 2002). In feature recognition of 3D faces, one has to take into consideration the obtained frontal view. Face recognition systems suffer drastic losses in performance when the face is not correctly oriented. The normalization process proposed here is a sequential procedure that aims to put the face shapes in a standard spatial position. Firstly, for the panning, the face is panned by angle which is defined by the mean and distance of the depth value of local areas that are divided into right and left, as defined in Eq. 11-14. For example, the given binary images presented in before compensation and after compensation for panning.

$$X_L = \frac{1}{n}\sum_{i=sl}^{sl-n} x_i \text{ and } X_R = \frac{1}{n}\sum_{i=s2}^{s2+n} x_i \tag{11}$$

$$P_L = \frac{1}{n}\sum_{i=sl}^{sl-3(max\_y-n)} \sum_{j=max\_y} D_{img'} if D_{img} > 0 \text{ and}$$
$$P_R = \frac{1}{n}\sum_{i=s2}^{s2-3(max\_y+n)} \sum_{j=max\_y} D_{img'} if D_{img} > 0 \tag{12}$$

$$L = X_L - X_R \text{ and } P = P_L - P_R \tag{13}$$

$$\theta = \tan^{-1}\left(\frac{P}{L}\right) \tag{14}$$

X_L: Centroid of left area, X_R: Centroid of right area, P_l: Mean value of left local area, P_M: Mean value of right local area,S1:max_x,θ: Roted angle.

Lee (2003) describe the transaction method secondly, for the rotation, the face is rotated by angle which is defined by the modified centroid and moments used in several area in Eq. 16. In general, centroid and moments are given as the equivalent of their continuous counterparts. Given B, which is the binary image, a set of n pixels (P_i Σ (x, y)) (i = 1,..., n), the coordinates (X_q, Y_q) of the centroid q of B are calculated by where p is contour line threshold values.

$$x_q^p = \frac{1}{n}\sum_{i=n}^{n} x_i^p \text{ and } y_q^p = \frac{1}{n}\sum_{i=n}^{n} y_i^p \tag{15}$$

Moments allow for a unique characterization of a shape. Given B, a set of n pixels (P_i Σ (x, y)) (i = 1,..., n) and (X_q, Y_q) its centroid, the definition of the discrete (k, l)-order central moment $\Pi_{k,l}^p$ of the set B is given by:

$$\Pi_{k,l}^p = \sum_{i=1}^{n} (x_i^p - x_q^p)^k (y_i^p - y_q^p)^l \tag{16}$$

A shape is uniquely represented by the set of all its (k, l)-ordered central moments. Except above, algorithm of the depth, area and angle of longitudinal section, volume of nose can be use (Lee, 2003). If compared conic spline and longitudinal methods, conic spline best because it useful for other checkpoints of face.

PIXALIZATION

The proposed algorithm based on physics base approaches for pixalization of an image, according to this method the first step consists in specifying the units along the x- and y- axes: if (x, y) are specified in terms of metric units (e.g., millimeters) and (x_s, y_s) are scaled versions that correspond to coordinate of the pixel, then the transformation can be described by a scaling matrix (Mathematics Book Publishing Committee, 1992).

$$\begin{bmatrix} x_s \\ y_s \end{bmatrix} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{17}$$

That depends on the size of the pixel (in metric units) along the x and y directions when $s_x = s_y$. Each pixel is square. In general, they can be different and then the pixel is rectangular. However here x_s and y_s are still specified relative to the principal points (where the z-axis intersects the image plane), whereas the pixel index (i, j) is conventionally specified relative to the upper-left corner and is indicated by positive numbers. Therefore, need to translate the origin of the reference frame to this corner

$$x' = x_s + o_x, \quad (18)$$

$$y' = y_s + o_y, \quad (19)$$

where, (O_x, O_y) are the coordinates (in pixel) of the principal point relative to the image reference frame. So the actual image coordinate are given by the vector $x' = [x, y, 1]^T$. Instead of the ideal image coordinate $x' = [x', y', 1]^T$. The above steps of coordinate transformation can be written in the homogeneous representation as:

$$x' = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & o_x \\ 0 & s_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (20)$$

where, x' and y' are actual image coordinates in pixels. In case the pixels are not rectangular, a more general form of the scaling matrix can be considered,

$$\begin{bmatrix} s_x & s_\theta \\ 0 & s_y \end{bmatrix} \in R^{2\times2} \quad (21)$$

where, S_θ is called a skew factor and is proportional to cot (θ) where, θ is angle between the image axes x_s and y_s. The transformation matrix in Eq. 3 then takes the general form

$$K_s = \begin{bmatrix} s_x & s_\theta & o_x \\ 0 & s_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \in R^{3\times3} \quad (22)$$

In many practical applications it is common to assume that $S_\theta = 0$.

Now combining the projection model with the scaling and translation yields a more realistic model of a transformation between homogeneous coordinates of its image expressed in terms of pixels,

$$\lambda \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & s_\theta & o_x \\ 0 & s_y & 0_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (23)$$

Notice that in the above equation, the effect of a real camera is in fact carried through two stages.

- The fist stage is a standard perspective projection with respect to a normalized coordinate system (as if the focal length were f = 1). This is characterized by the standard projection matrix$\Pi_0 = [I, 0]$.
- The second stage is an additional transformation (on the obtained image x) that depends on parameters of the camera such as the focal length f, the scaling factors s_x,s_y and S_θ and the centre offsets o_x, o_y.

The second transformation is obviously characterized by the combination of the two matrices K_s and K_f:

$$K = K_s K_f = \begin{bmatrix} s_x & s_\theta & o_x \\ 0 & s_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} fs_x & fs_\theta & o_x \\ 0 & fs_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \quad (24)$$

The coupling of K_s and K_f allows us to write the projection equation in the following way

$$\lambda x' = K\Pi_0 X = \begin{bmatrix} fs_x & fs_\theta & o_x \\ 0 & fs_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (25)$$

The constant 3×4 matrix Π_0 represents the perspective projection. The upper triangular 3×3 matrix K collects all parameters that are intrinsic to a particular camera and is therefore called the intrinsic parameter matrix, the calibration matrix of the camera. The entries of the matrix K have the following geometric interpretation:

O_x: x-coordinate of the principal point in pixels, O_y: y-coordinate of the principal point in pixels,

$fs_K = \alpha_x$ Size of unit length in horizontal pixels, $fs_y = \alpha_y$: Size of unit length in vertical pixels, α_x/α_y: aspect ratio σ,

fs_θ: Skew of the pixel, often close to zero.

Note that the height of the pixel is not necessarily identical to its width unless the aspect ratio $\sigma = 1$. When the calibration matrix K is known, the calibrated coordinates x can be obtained from the pixel coordinate x' by a simple inversion of K:

$$\lambda x = \lambda K^{-1} x' = \Pi_0 X = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (26)$$

The information about the matrix K can be obtained through the process of camera calibration. With the effect of K compensated for Eq. 24. Expressed in the normalized coordinate system, corresponds to the ideal pinhole camera model with the image plane located in front of the centre of the projection and the focal length f = 1. To summarize the geometric relationship between a point of coordinates $X_0 = [X_0, Y_0, Z_0, 1]^T$ relative to the world frame and its corresponding image coordinate $x' = [x', y', 1]^T$

(in pixel) depends on the rigid body motion (R, T) between the world frame and the camera frame (sometimes referred to as the extrinsic calibration parameters), an ideal projection Π_0 and the camera intrinsic parameters K. the overall model for image formation is therefore captured by the following equation:

$$\lambda\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} fs_x & fs_\theta & o_x \\ 0 & fs_y & o_y \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix}\begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \\ 1 \end{bmatrix} \quad (27)$$

In the matrix form, write

$$\lambda x' = K\Pi_0 X = K\Pi_{0g} X_0, \quad (28)$$

Or equivalently

$$\lambda x' = K\Pi_0 X = [KR, \quad KT] X_0, \quad (29)$$

Often for convenience call the 3×4 matrix $K\Pi_{0g}$ = [KR, KT] a (general) projection matrix Π, to be distinguished from the standard projection matrix Π_0.

Hence the above equation can be simply written as:

$$\lambda x' = K\Pi_0 X = [KR, \quad KT] X_0, \quad (30)$$

Compared to the ideal camera model the only change here is the standard projection matrix Π_0 being replaced by a general one Π.

At this stage in order to explicitly see the nonlinear nature of the perspective projection equation, can divide equation by the scale λ and obtain the following expressions for the image coordinates (x', y', z'),

$$x' = \frac{\Pi_1^T X_0}{\Pi_3^T X_0}, y' = \frac{\Pi_2^T X_0}{\Pi_3^T X_0}, z' = 1 \quad (31)$$

where, $\Pi_1^T, \Pi_3^T, \Pi_2^T, \Pi_3^T \in R^4$ are the three rows of the projection matrix Π.

PROPOSED ALGORITHM

In this study, proposed novel method of 3D face recognition can be stated in following steps –

Input :- Front face, side face. Output : - 3D animatable face, 3D realistic face.

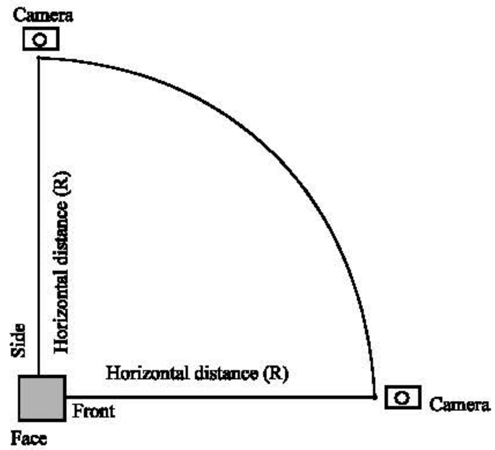

Fig. 3: Taking pictures, i.e., front and side view

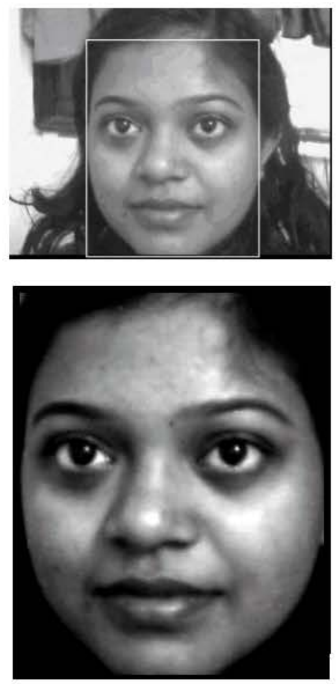

Fig. 4: Face extraction

Step 1: Take two views of pictures –front view on (x, y) and side view on (y, z) Fig. 3.

Step 2: Face Extraction. The face is extract from the front view in such a manner that the vertical dimension of the face from the front view and side view should be same and horizontal dimension of side view should cover the face, Fig. 4.

(Vertical dimension (Front face = Side view)).

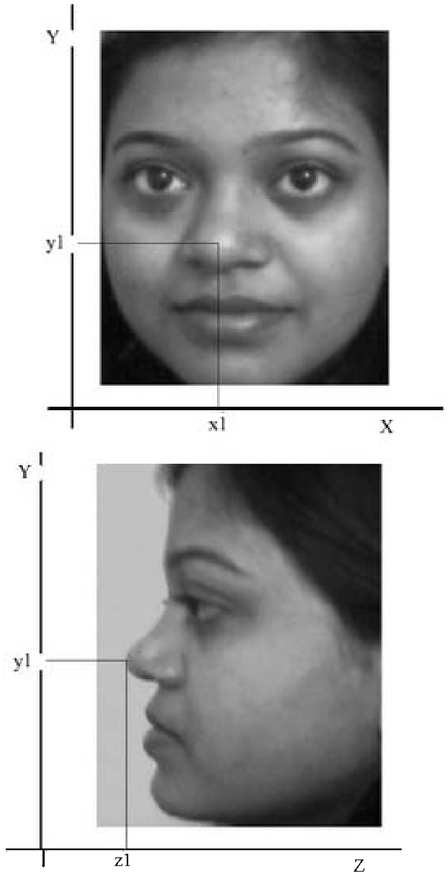

Fig. 5: Common point (vertical dimension)

Step 3: Condition If (coordinate of front view (x, y) = side view (y, z)) then combine this coordinates, we get coordinate of (x, y, z) of 3D Face Fig. 5.

(x, y) and (y, z) =(x, y z).
Go to 12 and 13

Step 4: If Compare (the each pixel to original face view) then (the extracted images of faces are then pixellized. This pixel indicating the points on images and each point have an identical coordinate from any reference line).

Apply procedure of pixalization from section 3.

Step 5: Print (3D Plotting of Coordinates (x, y, z))

Plotting 3D graph using 3d software of coordinate.

Step 6: Else If (x, y)>=(y, z) and (x, y) <=(y, z) then get Animatable 3d face and 3D Plotting of Coordinates (x, y, z).

Step 7: Go to step 10.(for check points)

Step 8: Print (get 3D image of face but this 3D face is not exact)

Step 9: Print (We modify above 3D face some check points (Eyes, Mouth and nose))
Apply algorithm for check points- Conic spline methods or depth area, angle of longitudinal section. Depth area and image of transaction and volume of nose.

Step 10: If (Check some points nose, eyes mouth) then use conic spline methods for this check points.

Step 11: Print (get 3D face view)

Step 12: If (3D face exact or not) then smoothing the 3D face.

Step 13: Print (3D Face Model-Finally it will give the 3D Face Model.)

Step 14: Stop process (Success)

CONCLUSION

In this study, proposed a new algorithm to automatic 3D Animatable face model from 2D image-through projection. The generated 3D facial models derived from two facial views. The above mentioned scheme is automatic and if compared with others this method will not be time consuming and being accurate. The proposed recognition algorithm is based on a projection. In future describe the next part of the method i.e., realistic 3D face. Of course our method doesn't give the realistic 3D face we modify through use of some methods of making 3D Animatable to 3D Realistic face. A new practical implementation of a person verification system using features of conic spline section and transaction and other facial, pixalization compensated 3D face image, is proposed. The eye interval and mouth width are also computed.

REFERENCES

Blanz, V. and T. Vetter, 2003. Face recognition based on fitting a 3D morphable model. IEEE. Trans. Pattern Anal. Mach. Intel., 25 (9): 1063-1074.

Bledsoe, W.W., 1966. Man-Machine Facial Recognition. Report on a Large-Scale Experiment. Technical Report PRI 22, Panoramic Research, Inc., Palo Alto, California.

Irfanoglu, M.O., B. Gokberk and L. Akarun, 2004. 3D shape-based face recognition using automatically registered facial surfaces. 17th International Conference on Pattern Recognition (ICPR'04), 4: 183-186.

Lee, Y.H., K.W. Park, J.C. Shim and T.H. Yi, 2002. 3D face recognition using projection vectors. Proceeding of IVCNZ2002, pp: 151-156.

Lee, Y., 2003. 3D Face recognition using longitudinal section and transaction. Proceeding of 7th Digital Image Computing: Techniques and Applications, 10-12 Dec.

Mathematics Book Publishing Committee, 1992. Linear Algebra and Geometry. Hyungseul Publishing Company.

Park, U., H. Chen and A.K. Jain, 2005. 3D model-assisted face recognition in video. Proceeding of 2nd Workshop on Face Processing in Video, British Colwnbia, Canada, May 8-11.